

# APPLICABLE MODEL TO PROCESS MANAGEMENT AGILE SOFTWARE DEVELOPMENT BASED ON CMMI 1.3 –DEV AND SCRUM

## MODELO APLICABLE A LA GESTIÓN DE PROCESOS ÁGILES DE DESARROLLO DE SOFTWARE BASADO EN CMMI –DEV 1.3 Y SCRUM

Recibido: 26 de agosto 2014- aceptado: 10 de diciembre 2014

**Adriana Iglesias<sup>1</sup>**  
Universidad Simón Bolívar

**Acellys Messino<sup>2</sup>**  
Universidad Simón Bolívar

**Patty Pedroza<sup>3</sup>**  
Universidad Simón Bolívar

**Ricardo Llanos<sup>4</sup>**  
Corporación Universitaria Americana

### Keywords:

SUMM, CMMI – DEV,  
SCRUM, Agile  
Management

### Abstract

This article of Research proposes a model oriented to the needs of total quality in organizations dedicated to software development according to the alignment of development and management processes with organizational objectives, which makes the formulation of relevant management model development process based on the principles and practices of agile approach towards the application of best practices and continuous improvement processes. The proposed model is called SUMM - "Unified Model Scrum Maturity" and is focused on process management software development agile framed in the best practices of CMMI-DEV 1.3 and SCRUM framework. SUMM consists of five maturity levels, according to the staged representation of CMMI - DEV, these levels in turn are composed of Goals, Objectives, Practices and indicators that enable the organization to employ an agile development model quality focused on continuous improvement processes.

### Palabras clave:

SUMM, CMMI – DEV,  
SCRUM, Agile  
MANAGEMENT

### Resumen

El presente Artículo de Investigación propone un modelo orientado a las necesidades de calidad total en las organizaciones dedicadas al desarrollo de software conforme a la alineación de los procesos de desarrollo y gestión con los objetivos organizacionales; lo cual hace pertinente la formulación de un modelo de gestión de procesos de desarrollo basado en los principios y prácticas del enfoque ágil, orientado hacia la aplicación de buenas prácticas y procesos de mejora continua. El modelo planteado se denomina SUMM - "Modelo Unificado de Madurez de Scrum" y está orientado a la gestión de procesos ágiles de desarrollo de software enmarcados en las buenas prácticas de CMMI-DEV 1.3 y el marco de trabajo de SCRUM. SUMM consta de cinco niveles de madurez, de acuerdo a la representación por etapas de CMMI - DEV, estos niveles a su vez están compuestos de Metas, Objetivos, Prácticas e Indicadores que permitirán a la organización poder emplear un modelo de desarrollo ágil de calidad, enfocados en procesos de mejora continua.

1. Universidad Simón Bolívar. Barranquilla Colombia. E-mail: aiglesias3@unisimonbolivar.edu.co
2. Barranquilla, Atlantico. E-mail: acellysm@gmail.com,
3. Universidad Simón Bolívar. Barranquilla, Colombia. E-mail: ppedroza1@unisimonbolivar.edu.co
4. Corporación Universitaria Americana, Servicio Nacional de Aprendizaje SENA. Barranquilla, Colombia. E-mail: castlellanos@gmail.com

## 1. INTRODUCCION

Las empresas desarrolladoras de software reconocen cada vez más la necesidad de la adopción de herramientas de ingeniería de software, procesos, metodologías, marcos de trabajo y modelos, que permitan establecer pautas y procesos al momento de desarrollar software, sin embargo solo reconocerlo no ha sido suficiente. Es de esta forma cómo conseguir desarrollar software con éxito es un desafío cada vez más exigente no solo porque las organizaciones requieren adquirir la capacidad de gestionar la complejidad, sino también la innovación tecnológica y el cambio de requisitos. Encontrar el equilibrio entre los valores de las metodologías ágiles y la robustez de los modelos tradicionales brinda una luz para dar frente a esta situación dado que la Gestión de la complejidad requiere disciplina en los procesos, mientras que la gestión del cambio exige capacidad de adaptación. Es así como, CMMI proporciona disciplina [1] y el marco de trabajo de Scrum mejora la adaptabilidad [2].

En este contexto es como se formula un modelo para abordar las problemáticas asociadas al proceso de gestión de desarrollo de software enmarcadas dentro de las metodologías ágiles y la formalidad de las buenas prácticas de CMMI – DEV en la feroz competencia del mercado vigente, en la que los productos de software quedan obsoletos rápidamente, y en donde los requerimientos de cortos tiempos de desarrollo, calidad en los procesos y productos, reducción de costos son cada En qué se está fallando?, este artículo se centrará en tres factores que han sido incidentes; en primera instancia se tiene que algunas organizaciones tratan de realizar un buen trabajo de desarrollo, sin contar con procesos bien definidos, esto se ha convertido en uno de los principales escollos debido a que éste proceso debe ser contemplado como un proceso de negocio que tiene que ser gestionado, ser eficiente, evolutivo, rápido y altamente adaptable [6]. Como segundo punto, muchas de éstas organizaciones de desarrollo de software aún no han implementado un modelo o metodología que gestione no sólo el proceso de desarrollo de software sino a toda la organización [7], esto se debe a factores como: desconocimiento del tema, temor a invertir en la implementación de una metodología de gestión del proceso de desarrollo de software al no ver la recuperación de la inversión a corto plazo o simplemente no han notado la importancia de valorizar sus proyectos de desarrollo en función de convertirse éstos en aporte a los logros de las metas del negocio. Y en tercera instancia la utilidad y eficacia de algunas prácticas, técnicas, metodologías y modelos varían entre unas organizaciones y otras, o unos proyectos u otros [8]. Lo anteriormente planteado en gran parte, se debe a que al mismo tiempo que unas prácticas son muy útiles para algunas

vez más frecuentes, pero para asumir estos retos, se hace necesario tener agilidad, complejidad, adaptabilidad y flexibilidad. [3]

## 2. CONTEXTO DEL PROBLEMA

En la actualidad la industria del software representa una importante actividad económica en muchos países del mundo, constituyéndose en una parte fundamental para el crecimiento de sus económicas. En países como Estados Unidos, Brasil, Canadá, India, China, Finlandia, entre otros, las pequeñas compañías de software que cuentan con menos de 50 empleados representan hasta un 85 por ciento de las compañías [2]. Sin embargo, cuando comienzan a crecer, en muchos casos, la falta de un proceso de desarrollo visible termina creando un caos en toda la empresa [3]. En esta búsqueda constante de adaptarse a procesos de desarrollo, metodología y/o marcos de trabajo las organizaciones, según la encuesta internacional denominada State of Agile Survey [4], tienen en cuenta factores como minimizar costos, optimizar tiempo, optimizar recursos y mejorar la calidad de sus productos.

Pero infortunadamente, Según el CHAOS Report [5], “Solo el 35% de los proyectos de software tienen éxito”, entonces

compañías, otras han sido consideradas muy negativamente por algunas de ellas.

Así, Turner y Jain [9] afirman que las organizaciones que usan métodos ágiles corren el riesgo de hacer demasiado énfasis en el conocimiento tácito y poco en la comunicación formal dentro de un equipo de desarrollo software, esto es en muchos casos dependiente de la experiencia de las personas que conforman el equipo, así como de su capacidad para compartir la información entre los diferentes miembros. Sin embargo, el proceso de desarrollo ágil de software no incluye únicamente comunicación informal, la comunicación formal: como código fuente, casos de pruebas, y una mínima cantidad de documentación es también utilizada en desarrollos ágiles, pero en la mayoría de los casos es deficiente [10], es en este punto en donde la aplicación de buenas prácticas como las de CMMI- DEV 1.3 [11] aportan al manejo del proceso de desarrollo en los ámbitos de pre-proyecto, proyecto y post-proyecto y transforman las debilidades de la informalidad de metodologías ágiles como SCRUM en fortalezas y nuevas oportunidades.

En el estudio realizado por Chow [7] sobre 109 proyectos ágiles se obtienen datos que evidencian que

existen prácticas que apuntan hacia el éxito o fracaso de un proyecto, Estos factores son: la estrategia de entrega, las técnicas ágiles de ingeniería de software (uso de estándares de codificación, refactorización, diseño simple, pruebas de integración), la capacidad del equipo, los procesos de gestión de proyecto, el entorno del equipo, y el compromiso del stakeholder y su vinculación en el proceso. En las siguientes tablas se puede observar un resumen de lo anteriormente mencionado:

Organización	Personas	Proceso	Técnicas
<ul style="list-style-type: none"> <li>• Soporte ejecutivo firme</li> <li>• Financiación, respaldo, gestión comprometida</li> <li>• Cultura organizacional cooperativa en lugar de jerárquica</li> <li>• Comunicación cara a cara</li> <li>• Entorno de trabajo apropiado</li> </ul>	<ul style="list-style-type: none"> <li>• Equipo con altas competencias y experiencia</li> <li>• Equipo con gran motivación</li> <li>• Jefes, gestores con conocimientos en procesos ágiles</li> <li>• Estilo de gestión flexible, adaptable</li> <li>• Equipos de trabajo auto-organizativos, coherentes</li> <li>• Buena relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Ámbito de proyecto no definido</li> <li>• Requerimientos de proyecto no definidos</li> <li>• Plan de proyecto no definido</li> <li>• Falta de mecanismos de trazabilidad del progreso del proyecto</li> <li>• Falta de presencia del cliente</li> <li>• No manejo de riesgos</li> <li>• Mala relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Estándares de codificación bien definidos</li> <li>• Actividades de refactorización</li> <li>• Correcta cantidad de documentación</li> <li>• Entrega de software periódica</li> <li>• Entrega de los principales requerimientos primero</li> <li>• Pruebas de integración</li> </ul>

**Tabla 1** Factores de éxito. Adaptada de [7].

Organización	Personas	Proceso	Técnicas
<ul style="list-style-type: none"> <li>• Falta de financiación o respaldo</li> <li>• Falta de gestión de los compromisos</li> <li>• Cultura organizacional demasiado tradicional</li> <li>• Cultura organizacional demasiado burocrática</li> <li>• Tamaño organizacional demasiado grande</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de habilidades necesarias</li> <li>• Falta de capacidad de gestión de proyecto</li> <li>• Falta de trabajo en equipo</li> <li>• Resistencia de grupos o individuos</li> </ul>	<ul style="list-style-type: none"> <li>• Ámbito de proyecto no definido</li> <li>• Requerimientos de proyecto no definidos</li> <li>• Plan de proyecto no definido</li> <li>• Falta de mecanismos de trazabilidad del progreso del proyecto</li> <li>• Falta de presencia del cliente</li> <li>• No manejo de riesgos</li> <li>• Mala relación con el cliente</li> </ul>	<ul style="list-style-type: none"> <li>• Falta de un conjunto de prácticas ágiles completo y correcto</li> <li>• Tecnologías y herramientas inapropiadas</li> </ul>

**Tabla 2** Factores de fracaso. Adaptada de [7].

Precisamente, para evitar los efectos negativos enunciados anteriormente, las empresas deben dedicar tiempo a la definición, adecuación y mejoramiento continuo de sus procesos organizacionales [12]. Con el fin de incrementar la probabilidad de finalizar un proyecto con éxito, cumplir con mayor precisión los tiempos planeados y los recursos invertidos o gastados, el desarrollo de soluciones estandarizadas y amigables al usuario, asegurando una mayor calidad del software y trazabilidad, facilitar el mantenimiento de las aplicaciones y simplificar la documentación.

A nivel nacional, en Colombia [13], se han realizado esfuerzos para mejorar la competitividad del sector informático, a través de la colaboración con modelos de calidad como CMMI, la capacitación comercial y administrativa, entre otros. Específicamente, respecto a los modelos de calidad, el sector público en conjunto con las empresas privadas ha realizado esfuerzos para mejorar la posición de las empresas de software

colombianas. El Gobierno Nacional, por medio de Proexport ha realizado un progreso sustancial en las empresas productoras de software con respecto a la evaluación de procesos, de acuerdo con el modelo CMMI; buscando así ampliar la base de dieciséis empresas Colombianas que están actualmente evaluadas según el SEI [14].

Sin embargo, la principal causa de la problemática realmente radica que en muchos casos estas empresas se mantienen al margen de las metodologías y los estándares internacionales debido a que tienen la percepción de que su implementación es difícil, costosa, y se consume mucho tiempo en el proceso [15].

Es así, como se pretende formular un modelo para abordar las problemáticas asociadas al proceso de gestión de desarrollo de software enmarcadas dentro de las metodologías ágiles y la formalidad de las buenas prácticas de CMMI-DEV en la feroz competencia del mercado vigente, en la que los productos de software quedan obsoletos rápidamente, y en donde los requerimientos de rapidez, calidad en los procesos y productos, reducción de costes son cada vez más frecuentes, pero para asumir estos retos, es necesario tener agilidad y flexibilidad.

### 3. MÉTODO

En primera instancia se realizó una revisión documental sobre los antecedentes asociados a las temáticas abordadas, partiendo de las metodologías ágiles de desarrollo de software [16] [17] [18] [19] [20] existentes hasta el modelo de buenas prácticas de calidad del software [21] [22] [23], a partir de los hallazgos encontrados en la revisión documental se planteó el modelo para la gestión de procesos ágiles de desarrollo de software, para lograrlo se abarcaron las siguientes fases metodológicas:

**Fase de Diagnóstico:** En esta fase se realizó un diagnóstico del estado en que son gestionados los proyectos de desarrollo de software, teniendo en cuenta parámetros esenciales relacionados con los factores claves que apunten hacia la implementación o no de metodologías, herramientas y modelos.

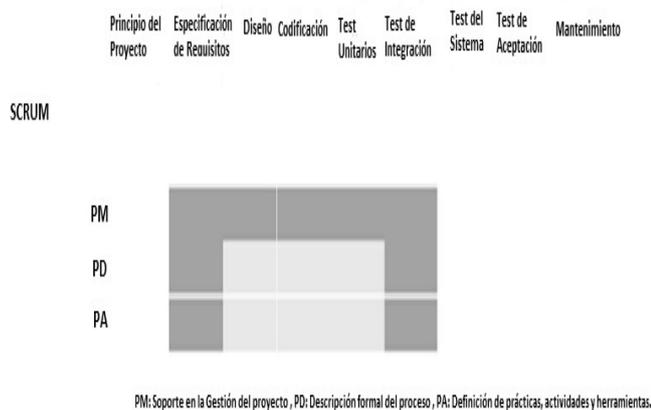
**Fase de Correspondencia:** En esta fase se realizó una caracterización del modelo de calidad escogido, CMMI – DEV 1.3, y el Marco de trabajo de Scrum. El objetivo de esta fase es indicar la correspondencia entre las áreas de procesos de CMMI y las prácticas de Scrum.

**Fase de Definición:** En esta fase se realizó la definición de las etapas y/o fases que conformaran el modelo.

Estableciendo los procesos inmersos e identificando los objetivos, metas, prácticas e indicadores.

#### 4. RESULTADOS

Una vez implementadas cada una de las fases metodológicas planteadas se obtuvieron los siguientes resultados que soportaron el modelo a partir del diagnóstico y de la caracterización realizada, de acuerdo a lo anterior se tiene que Scrum es un Marco de trabajo de gestión de proyectos, pero No presenta prácticas concretas que aseguren la calidad, su propósito no es ese, sino sacar el proyecto adelante de una forma colectiva y colaborativa [19] [20]. Tal y como se puede describir en la siguiente figura, en la cual se observa cómo se comporta cada una de las etapas de desarrollo de software en el marco de trabajo de Scrum.



**Figura 1. Scrum: áreas de proceso de desarrollo vs etapas de desarrollo.**

En la figura anterior se visualiza como actúan las áreas de proceso de desarrollo de software a lo largo de cada etapa en SCRUM. Se observa que este marco de trabajo en sus etapas de diseño, codificación y test de integración utiliza soporte en la gestión de proyecto y presentan deficiencia en los métodos que utilizan para describir sus prácticas, actividades y artefactos, adicionalmente no presenta una descripción formal del proceso al inicio del proyecto y en las etapas finales. Pero, ¿Por qué utilizar Scrum para el modelo planteado (SUMM)? [24] [25] [26] [27] [28], la siguiente presentación de criterios ayudará a entender por qué se hizo la selección de este marco de trabajo para el modelo propuesto. De igual forma, después de realizar un análisis se observa porque a partir de las falencias mostradas en la figura anterior se confirma que el marco de trabajo ideal para soportar el modelo propuesto es Scrum.

Metodología	Agile Management (APM)	Crystal Methods	Dynamic Systems Development Methods (DSDM)	Scrum	Test Driven Development	Extreme Programming (XP)
Define prácticas y actividades en cada etapa		X	X	X	X	
Integración con el cliente	X	X	X	X	X	X
Planificación de Requisitos	X	X	X	X	X	
Gestión de Calidad en el producto						
Integración de control de versiones				X		
Versiones integradas y planificación de iteraciones				X		
Acepta cambio constantes de requisitos				X		
Menos de 10 integrantes del equipo	X	X	X	X	X	X

**Tabla 3. Comparación de metodologías vs criterios de calidad**

De la comparación anterior se puede afirmar que aunque varias de las metodologías comparadas cumplen con los criterios analizados, Scrum presenta ventajas en varios de los criterios, entre los cuales se destacan:

- Fácil usabilidad, flexibilidad y adaptación mediante aceptación de requerimientos cambiantes.
- Los clientes se convierten en parte del equipo de desarrollo.

Se desarrollan planes de riesgos y mitigación frecuentes por parte del equipo de desarrollo, la mitigación de riesgos, la monitorización y la gestión de riesgos se lleva a cabo en todas las etapas.

La gestión de las expectativas del cliente a lo largo de todas las etapas del proyecto mediante: lista de requisitos priorizada y demostración de los resultados en cada iteración.

Resultados Anticipados por la utilización de listados priorizados de requisitos.

En cuanto a las desventajas de Scrum se observó que:

- Al principio del proyecto y en la especificación de requisitos no utiliza gestión del proyecto.
- En la etapa de finalización (test de aceptación, test del sistema y mantenimiento) no emplea gestión de proyectos.
- No tiene definido prácticas y actividades en las etapas del principio del proyecto

Una vez obtenidos los resultados del diagnóstico y haber identificado las ventajas y los campos de acción de Scrum sobre otras metodologías ágiles, se procedió a realizar una correspondencia entre las prácticas de Scrum, de acuerdo al manifiesto ágil y las veintidós (22) áreas de procesos de CMMI – DEV 1.3. [29][30][31]. En las siguientes tablas se condensa la comparación de acuerdo a la representación escalonada o por etapas de CMMI

**Correspondencia Scrum y CMMI – DEV Nivel 2**

• **Gestión de requerimientos - REQM**

En esta área de proceso el propósito es producir y analizar los requerimientos del cliente, de producto y de los componentes del producto.

CODIGO	PRACTICA CMMI	SCRUM
SP 1.1	Obtener las necesidades, las expectativas, las restricciones, y las interfaces de las partes interesadas para todas las fases del ciclo de vida del producto.	Revisar los requisitos con el propietario del producto y las partes interesadas.
SP 1.2	Obtener el compromiso de los participantes de proyecto sobre los requerimientos.	Dar a conocer las sesiones de planificación de las iteraciones para buscar compromiso por parte del equipo.
SP 1.3	Gestionar los cambios a los requerimientos a medida que evolucionan durante el proyecto.	1. Agregar los cambios de requisitos en la lista de requisitos del producto. 2. Gestionar los cambios en la reunión de planificación de la próxima iteración.
SP 1.5	Identificar las inconsistencias entre los planes del proyecto, los productos de trabajo y los requerimientos.	1. Establecer reuniones diarias para identificar los problemas. 2. Dar a conocer las sesiones de planificación para hacer frente a los problemas. 3. Tabla de seguimiento a las iteraciones para estimar el esfuerzo restante.

**Tabla 4. Correspondencia Práctica CMMI Vs Scrum En REQM.**

• **Planificación de proyectos – PP**

En términos básicos la finalidad de Planificación de proyecto (PP) es establecer y mantener planes que definan las actividades del proyecto.

CODIGO	PRACTICA EN CMMI	SCRUM
SP 1.1	Establecer una estructura de descomposición del trabajo (WBS) de alto nivel para estimar el alcance del proyecto.	Establecer la lista de objetivos/requisitos priorizada.
SP 1.2	Establecer y mantener las estimaciones de los atributos de los productos de trabajo y de las tareas.	La lista de objetivos/requisitos deben ser expresadas en forma de historia de usuarios. . Para cada objetivo/requisito se indica el valor que aporta al cliente y el coste estimado de completarlo.
SP 1.3	Definir las fases del ciclo de vida del proyecto en las que encuadrar el esfuerzo de la planificación.	Proceso definido por scrum.
SP 1.4	Estimar el esfuerzo y el coste del proyecto para los productos de trabajo y para las tareas, basándose en estimaciones razonadas.	Estimación de tiempo ideal. En la lista de objetivos/requisito se indican las posibles iteraciones y las entregas (releases) esperadas por el cliente .

**Tabla 5. Correspondencia prácticas CMMI vs Scrum en PP.**

• **Monitoreo y control del proyecto**

El propósito es proporcionar una comprensión de los avances del proyecto de manera que las acciones correctivas apropiadas pueden ser tomadas cuando los

resultados del proyecto se desvían significativamente del plan.

CODIGO	PRACTICA CMMI	PRACTICA SCRUM
SP 1.1	Monitorear los valores actuales del proyecto de acuerdo al plan	• Sprint burndown chart . • Release burndown chart realiza el seguimiento de los story points • Project Task Board, es usado para realizar los seguimientos de los Stories
SP 1.2	Seguimiento de los compromisos identificados en el plan del proyecto..	• Reunión Daily Scrum • Reunión Sprint review • Sprint burndown chart • Release burndown chart Rastrea los Story points que han sido completados
SP 1.5	Monitoreo de los involucrados según el plan del proyecto	• Reunión Daily Scrum • Reunión Sprint review
SP 1.6	Revisar periódicamente el progreso del proyecto, el rendimiento y Otros asuntos	• Reunión Daily Scrum • Reunión Sprint review

**TABLA 6: Correspondencia Prácticas CMMI VS SCRUM En Monitoreo Y Control.**

• **Gestión de la configuración**

No se especifica Ninguna práctica asociada en Scrum. Teniendo en cuenta que la Gestión de la configuración, requiere algo más que un control de versiones.

• **Aseguramiento de la Calidad de Producto y Proceso - PPQA**

Scrum no implica automáticamente la aplicación PPQA. A pesar de que Algunas actividades básicas de la PPQA se están haciendo naturalmente cuando el Scrum Master comprueba que el Proceso de Scrum se está siguiendo:

- comentarear código,
- revisión de documento,
- pruebas,
- eliminación de procesos y obstáculos,
- ineficiencia

Sin embargo NO Formaliza específicamente

- Gestión de los niveles de Procesos Objetivos
- Estándares y procesos que deben ser definidos y usados
- **Medición y Análisis – MA**

El propósito es desarrollar y mantener una capacidad de medición que se utiliza para apoyar la información de gestión necesidades. No hay prácticas en Scrum que establezcan un programa de medición similar al de MA. Sin embargo, las medidas en Scrum se pueden utilizar para implementar MA.

## Correspondencia Scrum y CMMI – DEV Nivel 3, 4 y 5

Los siguientes componentes no son fácilmente implementados por Scrum:

- Enfoque del Proceso Organizacional
- Definición de procesos organizacionales
- Formación Organizacional
- Gestión Integrada de Proyectos
- Gestión de Riesgos
- Análisis de Decisiones y Resolución
- Verificación
- Validación

Partiendo de esta correspondencia se puede entender que Scrum es el complemento adecuado combinada CMMI en el híbrido del modelo que se plantea (SUMM).

### Modelo Unificado de Madurez de SCRUM – SUMM

El modelo propuesto es un modelo híbrido para desarrollo de proyectos de software ágiles teniendo en cuenta aspectos inherentes a la calidad del software, este modelo denominado SUMM combina las prácticas de SCRUM y la robustez de CMMI - DEV en su versión 1.3., lo cual lo transforma en un híbrido entre lo tradicional y lo ágil.

SUMM, es la sigla que describe el “Modelo de Madurez Unificado de Scrum”, este consta de cinco (5) niveles los cuales son: Inicial, Gestionado, Definido, Mejorado y Optimizado. A través de esta representación se busca que una organización se enfoque en mejorar un conjunto de procesos relacionados obteniendo así un nivel de madurez enmarcado en procesos de mejora continua, el diseño de SUMM por su misma naturaleza híbrida está basado en la representación por etapas de CMMI.

SUMM brinda un enfoque estructurado orientado a la mejora de los procesos de Scrum paso a paso, la consecución de una etapa sugiere que se ha dado un mejoramiento, de igual manera, se habrá establecido una base para dar inicio a la siguiente etapa. Entre las características de SUMM se encuentran que este modelo aplica a proyectos de desarrollo de software ya que está cimentada en CMMI – DEV, por estar basada en los principios y prácticas de Scrum los proyectos deben desarrollarse en un lapso no mayor a 6 meses, de igual forma los equipos de desarrollo deben estar conformados con un número máximo de 10 integrantes sin contar a los usuarios ni al cliente.

En la siguiente figura se observa la representación de SUMM propuesta, la cual consta de un nivel inicial uno (1) y el máximo nivel de madurez que es el cinco (5).

**Figura 2.** Niveles de Madurez representación por etapas SUMM



### Niveles De Madurez Modelo Unificado de Madurez de Scrum – SUMM

SUMM contempla cinco (5) niveles de madurez, asociados a cada nivel se establecen metas basadas en las prácticas, principios y valores del marco de trabajo de Scrum.

Los niveles de SUMM están compuestos por Metas las cuales son un conjunto de prácticas que se definen con el fin de alcanzar determinados Objetivos. A continuación se presentan la relación de Metas, Objetivos y Prácticas de SUMM de manera general.

Metas	Nivel de Madurez	Objetivos	Prácticas
Gestión del proceso de SUMM	2	4	18
Gestión de Requisitos	2	3	12
Gestión de la comunicación con los Stakeholders	3	2	6
Validación y Verificación SUMM	3	1	5
Gestión de Sprint	3	3	13
Gestión cuantitativa de proyectos	4	1	2
Rendimiento de los procesos organizacionales	4	1	2
Análisis de causas y resolución SUMM	5	1	4
<b>Totales</b>		<b>16</b>	<b>62</b>

**Tabla 7.** Niveles de Madurez Metas, Objetivos y Prácticas

De acuerdo a la anterior tabla, SUMM está compuesto de Ocho (8) Metas que componen el marco general de cada nivel de madurez, de igual forma, lo conforman un total de dieciséis (16) Objetivos y Sesenta y dos Prácticas (62) que darán cumplimiento a cada Meta. Cada nivel de madurez en SUMM se mide por el alcance de los Objetivos determinados por cada Meta o por un conjunto de ellas. Para tener una visión global de cada uno de los niveles de madurez anteriormente descritos respecto a su representación

A continuación se describe cada nivel de Madurez de SUMM

**Nivel Inicial:** El nivel Inicial no consta de una estructuración de procesos, de igual manera, no hay objetivos que apunten hacia una mejora de procesos definidos, es el nivel más bajo de madurez dentro de SUMM. En este nivel, se encuentran las organizaciones que emplean algunas prácticas del marco de trabajo de Scrum, pero que aún no han definido con toda la formalidad y rigurosidad el cumplimiento del proceso de desarrollo bajo el marco de trabajo, ni mejoras de procesos. Principalmente, los problemas que se pueden asociar en este nivel son: incumplimiento en los tiempos proyectados, dificultades en la comunicación fluida entre el Team y los Stakeholders, calidad del software, costos de desarrollo.

El éxito en los proyectos de estas organizaciones generalmente depende de la capacidad y el nivel de preparación y competencia de su personal. Las organizaciones que no cumplan con las metas definidas para el nivel 2 están clasificadas en el nivel 1.

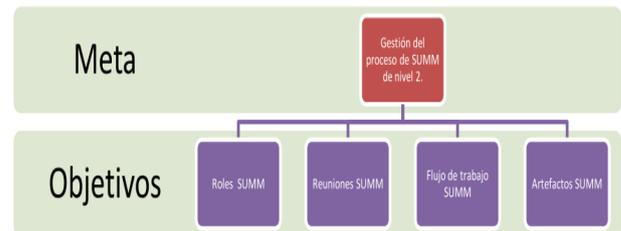
### Nivel Gestionado

En este nivel los procesos asociados a las prácticas de desarrollo de software bajo el marco de trabajo de Scrum son más estructurados y completos que en el nivel anterior. Los objetivos de este nivel son principalmente la planificación del proyecto, Mejorar la gestión de requisitos, Mejorar las prácticas de valor, colaboración y planificación de procesos. Los problemas asociados a este nivel son aquellos relacionados con la comunicación de los stakeholders, establecimiento de estándares de codificación e integración y la satisfacción del cliente.

En este nivel la organización debe establecer las pautas de la planeación de proyectos a través del planning poker u otros métodos de planeación definidos por el marco de trabajo, se entiende planning poker como una técnica que ayuda a todo el equipo de Scrum a realizar estimaciones basados en el consenso, es empleado para estimar el esfuerzo o el tamaño relativo de las tareas del proceso de desarrollo de software descritas en el sprint backlog. Por otra parte, El objetivo de este nivel, es ayudar al Team y a los Stakeholders a identificar y mejorar los problemas relacionados con la planificación, ingeniería de requisitos en lugar de solo aprender de los éxitos y fracasos del proyecto anterior, estableciendo una bitácora de lecciones aprendidas. Para dar cumplimiento al objetivo de este nivel, se propone dar cumplimiento a dos (2) metas las cuales son Gestión del proceso de SUMM y Gestión de Requerimientos.

**Gestión del proceso de SUMM:** Esta meta tiene como propósito el cumplimiento del flujo básico del marco de trabajo de Scrum, estableciendo roles, división de la organización, en equipos pequeños, interdisciplinarios y auto-organizados, división del trabajo, en una lista de

entregables pequeños concretos y priorizados, división del tiempo, en Sprint no superior a 4 semanas, generando código potencialmente entregable y testeado al finalizar cada sprint, plan de entregas, actualizando prioridades en colaboración con el stakeholder, Sin embargo, en este nivel las organizaciones a pesar de que gestionan sus procesos de desarrollo a través del marco de trabajo de Scrum, de acuerdo a las prácticas mencionadas anteriormente, es posible que a dichas prácticas no se



les dé cumplimiento de manera eficiente, eficaz y óptima.

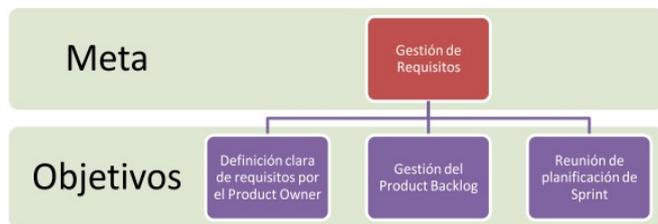
A continuación, se podrá visualizar la estructura general de la Gestión del proceso de Scrum de nivel 2.

**Figura 3,** Estructura General Gestión Proceso Scrum Nivel 2 de SUMM.

La estructura general de la meta de Gestión de Proceso de SUMM en el nivel 2 de SUMM describe cuatro (4) objetivos pilares los cuales son: Asignación de roles de SUMM, Gestión de reuniones de SUMM, Flujo de trabajo de SUMM y los artefactos de SUMM, como se visualiza en la anterior figura.

**Gestión de Requisitos:** Mientras que se realiza la planificación, se estiman la duración de los sprint y se organiza la pila de producto de acuerdo a las necesidades planteadas por el stakeholder a través del SUMM Product Owner del proyecto, se deben tener en cuenta los requisitos para poder establecer los compromisos de los participantes del proyecto, del mismo modo se debe comparar los requisitos de acuerdo al alcance y la definición misma del proyecto para poder identificar cualquier inconsistencia, incongruencia y evitar las ambigüedades. Según lo anterior de manera general, la Gestión de requisitos comprende un conjunto de prácticas que las organizaciones de nivel 2 de SUMM deben cumplir con el fin de lograr la satisfacción de la calidad del producto final desarrollado por la organización teniendo en cuenta la calidad del proceso para concebirlo.

A continuación, se podrá visualizar la estructura general de la Gestión de requisitos de nivel 2 de SUMM.



**Figura 4.** Gestión de requisitos de nivel 2 de SUMM.

Como se ilustra en la figura anterior, la gestión de requisitos de nivel 2 de SUMM, destaca la necesidad de una definición clara de requisitos por parte del SUMM Product Owner para dirigir el SUMM Team y la correcta aplicación de los conceptos de Scrum en relación a la gestión del Product Backlog y reuniones de planificación de Sprint exitosas.

En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.

### Nivel Definido

Definido, corresponde al nivel 3 de SUMM, en este la relación con los stakeholders, los procesos de validación y verificación, las entregas a tiempo según lo planeado son el pilar principal. De acuerdo a lo anteriormente planteado, las metas de este nivel son la Gestión de relaciones con el cliente, comunicación SUMM Team y la Gestión de sprint.

Las organizaciones en este nivel, tienen un mayor control sobre las prácticas relacionadas con el establecimiento de canales efectivos de comunicación con los stakeholders, la gestión de riesgos pasa a ser un ítem importante, sin embargo, no se realiza un panorama de riesgos de forma rigurosa ni exhaustivo.

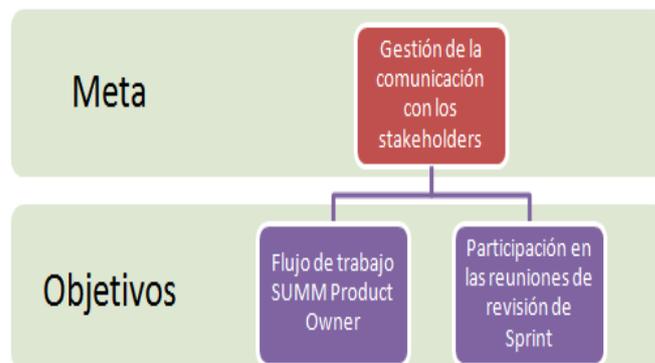
Para dar cumplimiento al objetivo de este nivel, se propone dar cumplimiento a cuatro (3) metas, las cuales son Gestión de la comunicación con los stakeholders, Validación y verificación SUMM y Gestión del Sprint

### Gestión de la comunicación con los stakeholders:

Si se parte del hecho de que el propósito principal en los procesos de desarrollo de software es la satisfacción de las necesidades de los stakeholders, se puede afirmar que estos son el pilar fundamental que brinda apoyo a estos procesos. En las metodologías ágiles los stakeholders pasan de ser un actor pasivo a participar activamente en el proceso de desarrollo, inclusive en Scrum estos ocupan uno de los roles dentro del flujo de trabajo como se ha evidenciado en ítems anteriores.

Basados en lo anterior, se hace necesario incrementar los esfuerzos en estrechar la comunicación con los

stakeholders, procurando establecer canales efectivos y eficaces entre los stakeholders, su representante, SUMM Product Owner, SUMM Master y el SUMM Team. A continuación, se podrá visualizar la estructura general de la Gestión de la comunicación con los stakeholders.



**Figura 5.** Estructura general Gestión de la comunicación con los Stakeholders SUMM nivel 3

Como se ilustra en la figura anterior, la gestión de la comunicación con los stakeholders de SUMM, es una meta importante ya que se informa a los stakeholders de una forma clara, transparente y veraz, cómo va el proceso. En los proyectos propiciados en entornos distribuidos, el SUMM Product Owner debe con la ayuda del SUMM Master colocar a disposición de los stakeholders los mecanismos de comunicación suficientes para que éstos puedan involucrarse en el proyecto. De esta manera, el SUMM Team puede gestionar el flujo trabajo en los Sprint planeados de la forma más eficaz posible, sorteando los obstáculos que se encuentren durante su desarrollo y ejecución, de tal forma que, tanto las funcionalidades como los requisitos no funcionales que han demandado, sean fielmente plasmados en el producto desarrollado.

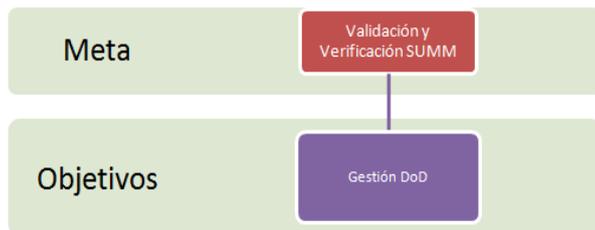
**Validación y verificación SUMM:** En los procesos de desarrollo de software se evidencia la necesidad de que los Stakeholders y sus representantes validen si el producto intermedio y/o final, dependiendo de cómo se haya descrito en el sprint Backlog, se ajusta con las necesidades implícitas y explícitas planteadas por ellos al SUMM Team.

De acuerdo con lo anterior, estas validaciones se deben realizar en los procesos de revisión de los Sprint ejecutados una vez se finalice este sobre el producto entregable funcional resultado del sprint. Una vez concluido los procesos de validación debe existir la retroalimentación por parte del Stakeholder o en su defecto del SUMM Product Owner. Obtenida la retroalimentación en la reunión retrospectiva el

SUMM Team en cabeza del SUMM Master estos deben analizar los resultados para tomar acciones correctivas, en caso de que se requiera.

Por otra parte, los procesos de verificación se orientan a comprobar que el software que se está desarrollando da cumplimiento a su especificación. Se comprueba que el sistema cumple los requerimientos funcionales y no funcionales que se le han especificado impuestos al inicio de cada Sprint, de acuerdo a lo descrito en el Product Backlog.

A continuación, se podrá visualizar la estructura general de la validación SUMM



**Figura 6.** Estructura general Validación y Verificación SUMM

Como se ilustra en la figura anterior, la validación y verificación SUMM, proporciona procesos de evaluación de productos, para conseguirlo se hace necesario la “Definición de Done” (DoD), si el SUMM Team cumple con los criterios definidos se podrá asegurar que se está cumpliendo con las necesidades del Stakeholder, no solo en términos de funcionalidad, sino también de calidad. La DoD consiste en definir una lista de actividades entre las cuales se encuentra: Buenas prácticas de codificación (código limpio), pruebas unitarias, pruebas de integración, documentos de diseño que se agregan valor verificable al producto.

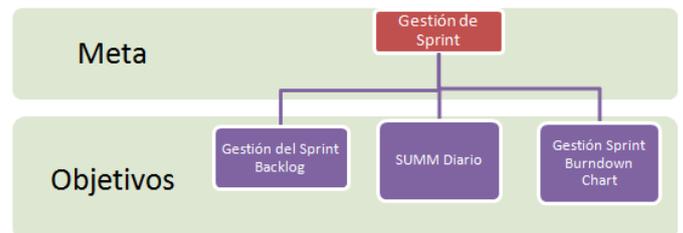
Sin embargo, se puede establecer DoD en varios niveles estos son: DoD para un Story establecido en el Product Backlog, DoD para Sprint y DoD para la liberación del producto, este último determina un estado de potencialmente entregable. Entre las características DoD se pueden destacar que este no es estático, tal y como el marco de trabajo tampoco lo es y que la DoD debe ser una lista de verificación auditable. En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.

**Gestión de Sprint:** La gestión de Sprint contribuye a aumentar los niveles de satisfacción del cliente a través de un conjunto de prácticas que deben cumplirse y aplicarse con el fin de administrar los procesos de entrega sus los proyectos desarrollados a tiempo y acorde a su presupuesto.

De igual manera, la gestión de Sprint ayuda al SUMM Team en la monitorización de su progreso, estableciéndose como el indicador que informará tanto al SUMM Master como al SUMM Team sus posibilidades de cumplir los Stories al finalizar el Sprint, respecto a los tiempos proyectados. Por otro lado, la gestión de Sprint también guía al SUMM Product Owner a analizar el nivel de avance del desarrollo respecto al producto.

En la Gestión de sprint es importante tener en cuenta que: la estimación de nuevas tareas y la reestimación de tareas que se encuentran en progreso requiere de un esfuerzo considerable, el esfuerzo asignado a las tareas a menudo suele ser impreciso, debido al mismo proceso de estimación, es importante tener en cuenta no estimar el sprint en unidades de tiempo sería mejor hacerlo en esfuerzos, el SUMM Team debe tener presente que la terminación de una tarea o conjunto de ellas, no entrega valor alguno; sólo los Stories completos entregan valor de acuerdo al Product Backlog definido.

A continuación, se podrá visualizar la estructura general de la Gestión de Sprint SUMM



**Figura 7.** Estructura general Gestión de Sprint SUMM.

Como se ilustra en la figura anterior, la Gestión de Sprint SUMM proporciona procesos para la administración de las iteraciones, para conseguirlo se hace necesaria la Gestión del Sprint Backlog, a través del flujo Diario SUMM y por último es importante establecer la medición de la velocidad del SUMM Team para dar cumplimiento a cada iteración por medio del Sprint burndown chart. En este orden de ideas, para alcanzar una meta se debe dar cumplimiento a los objetivos descritos para esto, adicionalmente a las prácticas asociadas y a las métricas de uso para medir la calidad de la implementación de estas.

Con la implementación del nivel 3 de SUMM, una organización puede tener éxito en el desarrollo de varios proyectos de software, sin embargo, este éxito es sólo parcial debido a la falta de una gestión estandarizada que garantice la misma calidad y rendimiento en todos los procesos de desarrollo.

## Nivel Mejorado

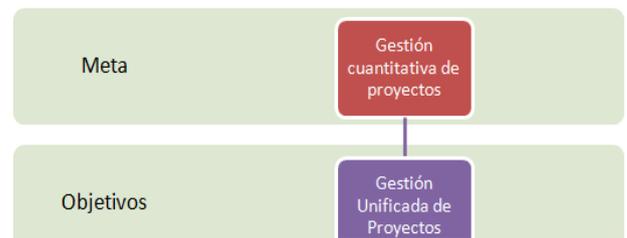
El Nivel Mejorado o también conocido como mejorado cuantitativamente, es el cuarto nivel en la representación de SUMM, en este las organizaciones poseen procesos de desarrollo de software y gestión de proyectos estandarizados apoyados en la gestión de rendimiento de los procesos a través de la medición y el análisis de las prácticas asociadas, esto se debe a que los procesos son entendidos y controlados cuantitativamente, es decir, mediante mediciones detalladas. Este nivel de madurez de SUMM, lo conforman dos (2) metas, las cuales son la Gestión cuantitativa de proyectos y el Rendimiento de los procesos organizacionales

**Gestión cuantitativa de proyectos:** esta meta conduce a las organizaciones a emplear el mismo proceso de desarrollo en todos los proyectos que ejecute, es decir, se estandariza y se maneja un lenguaje común en la organización respecto a los procesos de desarrollo. Lograrlo no es una tarea sencilla, ya que la organización debe completar un proceso de normalización, si se parte del hecho de que existen elementos comunes a los procesos de desarrollo de proyectos, entre los cuales están, herramientas, metas, objetivos, prácticas y que todos los proyectos tienen como punto de partida la planeación inicial, se pueden realizar estimaciones en pro de evitar errores y optimizar el proceso mediante acciones de mejora continua. Dichas estimaciones se pueden calcular basadas en métricas establecidas por la organización.

La gestión cuantitativa de proyectos aporta a las organizaciones: garantía de calidad en el proceso de desarrollo, una definición y mejora en los proyectos ejecutados, al SUMM Team y al SUMM Master aporta conocimiento mejor control del esfuerzo estimados, el seguimiento y divulgación de problema por medio de las lecciones aprendidas.

Teniendo en cuenta lo anteriormente mencionado, la gestión cuantitativa puede predecir el grado en el cual el proyecto podría satisfacer sus objetivos de calidad y funcionalidades explícitas e implícitas, teniendo en cuenta información estadística de proyectos anteriores, realizando una especie de "análisis post-mortem". De igual forma, el desarrollo de proyectos de software abarca varios procesos de la organización, estos procesos deben ser identificados claramente teniendo en cuenta los subprocesos asociados, tener esto claro será de gran ayuda para lograr un control en cuanto a calidad y funcionamiento.

A continuación, se podrá visualizar la estructura general de la Gestión cuantitativa de proyectos nivel 4 de SUMM.



**Figura 8.** Gestión cuantitativa de proyectos nivel 4 de SUMM

**Rendimiento de los procesos organizacionales:** La principal característica de esta meta es establecer la interpretación cuantitativa de los resultados obtenidos a partir de la estandarización de los procesos de la organización asociados a la gestión de proyectos de desarrollo de software con el objeto de obtener un mayor rendimiento de los procesos y proporcionar información para la Gestión Cuantitativa de Proyectos.

Este exige el seguimiento de todas las prácticas sugeridas hasta el nivel cuatro de madurez SUMM. Lo anterior permitirá, proporcionar información suficiente sobre el proceso real y gestionar el rendimiento, determinar si los procesos son predecibles, identificar si los procesos en el rendimiento están dentro de los límites y son coherentes, establecer criterios que permitan determinar las medidas y las técnicas de análisis que emplearan en procesos y subprocesos, identificar los procesos inusuales, identificar los procesos que pueden ser estandarizados.

A continuación, se podrá visualizar la estructura general del Rendimiento de los procesos organizacionales nivel 4 de SUMM.



**Figura 9.** Rendimiento de los procesos organizacionales nivel 4 de SUMM

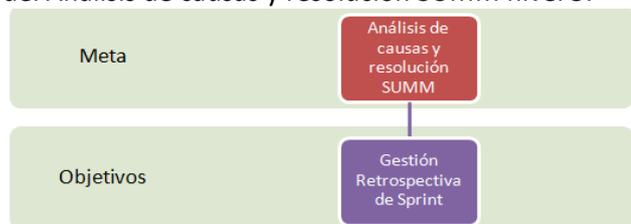
## Nivel Optimizado

En el nivel 5 de SUMM se busca una superación continua de la competencia por medio del aumento en el nivel de satisfacción de los Stakeholders e interesados y el SUMM Team. En este nivel, las organizaciones están obligadas a aplicar procesos de mejora continua como eje fundamental de su cultura organizacional. De igual manera, es en este nivel en el cual las organizaciones se brindan espacios para la innovación generando ideas o tecnologías que

otorguen ventajas comparativas y competitivas, Aunque en este último aspecto aún no se pudo establecer correspondencia solo hay una meta asociada a este nivel

**Análisis de causas y resolución:** esta meta tiene como pilar principal la identificación de las causas de los resultados escogidos, buscando mejorar el desarrollo de los procesos optimizándolo tanto de nivel de proyectos como a nivel organizacional. Para dar cumplimiento a esta meta se hace necesario un entendimiento cuantitativo del proceso para poder ser efectivas, como la mayoría de metas de nivel 4 y 5.

A continuación, se podrá visualizar la estructura general del Análisis de causas y resolución SUMM nivel 5.



**Figura 10.** Análisis de causas y resolución SUMM nivel 5

## 5. DISCUSION

A pesar de los múltiples esfuerzos, tendencias, metodologías, modelos, estándares, estrategias las cifras correspondientes a los niveles de fracaso y cancelación de proyectos sigue siendo amplia frente a los proyectos que logran culminar de manera exitosa. El mercado de desarrollo de software cada vez es más extenso y los clientes cada vez más demandan mejores desarrollos de acuerdo a sus necesidades con tecnología de punta, cortos tiempos de desarrollo, costos reducidos, altamente confiables y con calidad.

De acuerdo a lo anterior, se centró la investigación en proponer un modelo que permitiera la ejecución de proyectos de desarrollo de software manteniendo la calidad en los procesos y en los productos, fue así como se inició un proceso de revisión del estado del arte a través del cual se pudo identificar cuales metodologías, modelos o marcos de trabajo de desarrollo de software cumplían los criterios de: definición de prácticas y actividades por etapas o niveles no de manera secuencial sino de forma iterativa e incremental, inmersión del cliente en el proceso de desarrollo, ingeniería de requisitos, desarrollos a mediano y largo plazo. Fue de esta manera como después de realizar una comparación de las metodologías, modelos y marcos de trabajo encontrados se resolvió que Scrum cumplía la mayoría de los criterios establecidos, de igual forma al revisar estadísticas de uso de metodología, Scrum resultó ser de las más usadas o al menos una de las más conocidas en la actualidad dentro del enfoque ágil.

De igual forma, gracias a la revisión del proceso de desarrollo de Scrum se vislumbró algunas falencias en cuanto al empleo de buenas prácticas orientadas a la calidad del proceso propias del marco de trabajo, fue de esta forma, como se decidió incorporar las buenas prácticas de CMMI – DEV en su más reciente versión en el modelo propuesto.

Una vez definidos los componentes base pilares del modelo, es decir, proceso de desarrollo ágil basado en Scrum y buenas prácticas de calidad basadas en CMMI – DEV 1.3. Se procedió a realizar una correspondencia entre las prácticas genéricas y específicas de las áreas de proceso de CMMI – DEV y las metas, artefactos y Objetivos del proceso de Scrum. Dicha correspondencia, sirvió de base para poder establecer los procesos en los cuales se debía hacer énfasis en el modelo a proponer.

El modelo propuesto se compone de cinco (5) niveles de madurez enfocados en la representación por etapas de CMMI, estos niveles en su orden son: Inicial, Gestionado, Definido, Mejorado, Optimizado; cada nivel lo componen metas, a su vez las metas se componen de Objetivos y asociados a ellos están las prácticas y los indicadores. La principal dificultad a la hora de proponer el modelo se relacionó al hecho de no cargar tanto el modelo propuesto, sacrificando la agilidad de Scrum, ni tampoco ir en contravía de los valores y principios que lo componen.

A pesar de que en la correspondencia se encontró que en los niveles superiores de CMMI había carencia de prácticas de Scrum asociadas, se logró definir metas, objetivos, prácticas e inclusive indicadores para estos niveles, procurando siempre no sacrificar la agilidad. Respecto a los procesos de Validación y Verificación en los cuales tampoco había aplicación aparente y clara de prácticas de Scrum se logró definir prácticas que modelan ambos procesos, el empleo del artefacto DoD se convirtió en pieza clave para esta labor.

A pesar de que se realizó un esfuerzo para llevar el proceso de Scrum a los niveles 4 y 5 de CMMI, aún existen áreas de proceso que podrían emplearse y que de hecho son necesarias, sin sacrificar la agilidad, estas áreas serían la gestión de la configuración, la gestión de riesgos y la innovación e implementación organizacionales.

La implementación del modelo propuesto en al menos 5 organizaciones piloto que retroalimenten el modelo

a partir del día a día de las empresas de desarrollo de software en todos los niveles.

Establecer un método de Valoración y evaluación (Appraisal) a las organizaciones que permita establecer el nivel de madurez en que se encuentran y como deben incorporar los procesos descritos en el modelo propuesto. Construcción de mapa de procesos completo de SUMM, identificando cómo implementarla, Construcción de un documento Construcción de un documento guía donde se presenten elementos adicionales como formatos, cálculo de indicadores y procedimientos

#### REFERENCIAS

- [1] Software Engineering Institute. CMMI for Development. Pittsburg: s.n., CMU/SEI-2006-TR-008. 2006.
- [2] K. Beck, Manifesto for Agile Software Development, 2001. <http://agilemanifesto.org/>.
- [3] G. Paula, D. Parada, Agilidad y disciplina del proceso de desarrollo de software para las Pequeñas y Medianas Empresas (PYMES) y las Cooperativas en Latinoamérica. Caso: Venezuela. VII Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento. 2008
- [4] W. Dave, Water-Scrum-Fall Is The Reality Of Agile For Most Organizations Today. 2012. Obtenido de [http://www.versionone.com/state\\_of\\_agile\\_development\\_survey/11/](http://www.versionone.com/state_of_agile_development_survey/11/)
- [5] Standish Group International. The CHAOS Report. 2012. Obtenido de <http://www.standishgroup.com>
- [6] Richardson, Ita and Gresse, Christiane. Why are small software organizations different? IEEE Computer Society. Vol. 24. ISSN 0740-7459. 2007
- [7] T. Chow, a. B. A survey study of critical success factors in agile software projects. J. Syst. Softw. 2009, pp 961-971.
- [8] O. Salo, P. Abrahamsson, Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum. 2008. Pág 58--64.
- [9] R. Turner, A. Jain, Agile Meets CMMI: Culture Clash or Common Cause? In Extreme Programming and Agile Methods — XP/Agile Universe. 2002. pág. 153-165.
- [10] P. Abrahamsson, O. Salo, J. Ronkainen, J. Warsta, Agile Software Development Methods: Review and Analysis. 2002. Pág 478.
- [11] CMMI for Development, Version 1.3. Improving processes for developing better products and services. 2010. CMU/SEI-2010-TR-033 ESC-TR-2010-033. <http://www.sei.cmu.edu>
- [12] A. S. Marçal, Estendendo o SCRUM segundo as Áreas de Processo de Gerenciamento de Projetos do CMMI. CLEI 2007: XXXIII Conferencia Latinoamericana de Informática, San José, Costa Rica. 2007. Pág 9-12
- [13] S. Gallardo, El TLC bajo la lente de Fedesoft. 91 Sistemas. [Revista]. ISSN 0120-5919. 2005. <http://www.acis.org.co/index.php?id=388>.
- [14] J. H. Canós, P. Letelier, M.C. Penadés, "Metodología Ágiles en el Desarrollo Software". VIII Jornadas de Ingeniería de Software y Bases de Datos. 2003
- [15] Y. Estrada, Aprendizaje y aplicación del CMMI-DEV en PYMES de software Colombianas. La experiencia RCCS. Gerenc. Tecnol. Inform, Vol. 9, N° 24, 2010. pp 57 – 76.
- [16] Beck, K., Beedle, M., Bennekum, A. v., Cockburn, A., Cunningham, W., Fowler, M., y Otros. Manifesto for Agile Software Development. 2001.
- [17] K. Beck, Extreme Programming Explained: Embrace Change. Addison- Wesley. 2000
- [18] K. Beck, Test Driven Development: By Example. Addison Wesley. 2002
- [19] K. Schwaber, The Enterprise and SCRUM. Microsoft Press, Washington. 2007
- [20] K. Schwaber, SCRUM Development Process Advanced Development Methods. 2007
- [21] CMMI Guía para la integración de procesos y la mejora de productos. Segunda edición. Pearson Educación. Cátedra de Mejora de Procesos de Software en el Espacio Iberoamericano de la Universidad Politécnica de Madrid. ISBN: 9788478290963. 2009.
- [22] P. Pedroza, Modelo para la gestión de la calidad de servicios TI Aplicable a las pymes de la ciudad de Barranquilla. Universidad del Norte. 2010
- [23] CMMI for SCAMPISM Class A Appraisal Results 2012 Mid-Year Update. Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213. 2012. <http://www.sei.cmu.edu/cmmi/results.html>
- [24] K. Orr, CMM versus agile development: Religious Wars and Software Development. Cutter Consortium. Executive Report. Vol.3 N° 7, 2002.
- [25] M. Paulk, Extreme Programming from a CMM Perspective, IEEE Software, vol. 18, no. 6, p.19-26, 2001.
- [26] J. Berrocal, J. García-Alonso, J. Murillo, Agilizando las herramientas de gestión de proyectos. Actas de 1ª Conferencia Agile-Spain CAS 2010. 2010
- [27] R. Casallas, H. Arboleda, QualDev Process: Procesos adaptables de desarrollo de software para proyectos ágiles. 2008
- [28] O. Salo, Abrahamsson, et.al. Agile methods in European embedded software development organizations: a survey on the actual use and usefulness of Extreme Programming and Scrum. Software, 2008 págs. 58-64.
- [29] A. S. Marçal et al. Mapping CMMI Project Management Process Areas to SCRUM Practices. 31st Annual Software Engineering Workshop, Loyola College, Baltimore, MD, USA, 6-8 .2007
- [30] A. S. Marçal et al. Blending Scrum practices and CMMI project management process areas.

Innovations in Systems and Software Engineering Journal,  
Volume 4, Number 1. Springer London. 2008

[31] J. Sutherland, R. JAKOBSEN, K. JOHNSON, Scrum and  
CMMI Level 5: The Magic Potion for Code Warriors. The  
12th annual European Systems and Software Engineering  
Process Group Conference EUROPEAN SEPG 2007,  
Amsterdam, 2007.